REMARKS

The amendments and remarks presented herein are consistent with those noted in the recent telephone call from Applicant's representative to the Examiner. Accordingly, entry of this amendment and reconsideration of the pending claims is respectfully requested.

The Office Action, mailed October 31, 2007, considered and rejected claims 1-12, 14-22, and 24-29. Claims 1-12, 14-22, and 24-29 were rejected under 35 U.S.C. § 103(a) as being unpatentable over CERT CC, "CERT Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests" (CERT-Advisory) in view of Cert CC, "Understanding Malicious Content Mitigation for Web Developers" (CERT) in view of Wheeler, Secure Programming for Linux and Unix HOWTO in view of *Sanin* (U.S. Publ. No. 2004/0073811).[1] The Specification and drawings were also objected to, which objections are moot in view of the amendments to the specification (mooting the objection to the specification) and the claims (mooting the objection to the drawings). Claims 4, 11 and 21 were also objected to, which objection is also moot inasmuch as the claims have been cancelled.[2]

By this paper, claims 1-29 are cancelled, and claims 30-45 added. Accordingly, following this paper, claims 30-45 are pending, of which claims 30, 44 and 45 are the only independent claims at issue.

While the support for each of the new claims is readily apparent from even a brief review of the original application, Applicant notes that support for the claim amendments can be found in at least paragraphs 7, 8, 15, 16, 18, 21, 22, 24-28, 30 and 31 or the originally filed application, as well as in the original claims and figures.

---

[1] Although the prior art status of the cited art is not being challenged at this time, Applicant reserves the right to challenge the prior art status of the cited art at any appropriate time, should it arise. Accordingly, any arguments and amendments made herein should not be construed as acquiescing to any prior art status of the cited art.

[2] Claims 1-12, 14-22 and 24-29 were also rejected under 35 U.S.C. §§ 101 and/or 112. Such rejections are also moot in view of the cancellation of such claims. Nevertheless, with respect to such rejections, Applicant continues to note that the Examiner has failed in his burden to provide a *prima facie* rejection. For example, with respect to the written description rejection, Applicant has, in each of the past actions, "specifically identified where the new and amended claims are supported. For example, footnote 5 on page 12 of Amendment D and footnote 3 on page 10 of Amendment C each clearly provide the specific paragraphs providing support for the claim amendments. Accordingly, Applicant continues to note that under M.P.E.P. 2164.04(I) the Examiner has failed in his burden as he has not provided any reasons why one of ordinary skill in the art would not have recognized the Applicant was in possession of the claimed subject matter at the time of the invention. Instead, the Examiner points to M.P.E.P. § 2163 and notes that the "examiner properly notes that the applicant has not pointed out where the new (or amended) claim is supported." Such assertion is clearly in error for at least the reasons presented above, including, but not limited to, the specific disclosure of Applicant reciting the precise paragraphs in which the support may be found.

As reflected in the above claim listing, Applicant's claims generally relate to methods and computer program products for mitigating cross-site scripting attacks of a third party against responses served from a web server to a user computer. As recited in claim 30, for example, an exemplary method includes receiving an HTTP request at the web server. The HTTP request was sent by the user computer and requests a response that includes text and HTML elements. Before the request is dynamically rendered, a script module of the server examines the HTTP request for script constructs identified in an updateable list of markers of active content that is stored at the web server. Such examination consists of examining only HTML elements where user input is introduced. A script construct is then found within a particular HTML element and, in response, an error is generated and the HTML request aborted. The user computer is then informed of the find and requested to resubmit a request. Claim 44 recites a similar method as being capable of performance due to is storage on a computer-readable medium of a computer program product. Claim 45 also recites a similar method, further adding that the list of script constructs is maintained at the web-server, and the finding of the script construct in a request for dynamic content in the form of an embedded link.

While *CERT-Advisory*, *CERT*, *Wheeler* and *Sanin* generally relate to preventing damage from attacks by third parties, Applicant respectfully submits that they fail to disclose or suggest Applicant's invention as reflected in the above claims. For example, among other things, the cited references fail to disclose or suggest wherein examining the HTTP request for script constructs consists of examining only HTML elements where user input is introduced, as recited in combination with the other claim elements. In other words, the cited references fail to disclose or suggest that when examining a request, the examination for script constructs is not made to additional elements beyond just the HTML elements where user input is introduced. Indeed, the cited references disclose and suggest that an entire request is examined.

As disclosed in *CERT-Advisory* and *CERT*, for example, a request received from a client is at least partially processed, even where it contains malicious code. For example, *CERT-Advisory* generally discusses the problem associated with malicious code from a cross-site scripting attack. (pp. 1-2). To address such problems, *CERT-Advisory* notes that web site developers can prevent such attacks by allowing only a limited character set or by filtering data during generation of the output page. (p. 5, ¶ 3-6). For additional details on encoding and filtering, however, *CERT-Advisory* refers to the *CERT* reference.

CERT adds to the discussion in *CERT-Advisory* regarding methods for avoiding damage due to cross-site scripting attacks. As explained in *CERT*, damage from such attacks can be minimized by filtering specific characters out of web pages that contain both text and HTML markup. (pp. 1, 4). For instance, a web page request may be filtered either during the data input or data output process to ensure that all dynamic content is filtered. (p. 4, ¶ 4). Thus, *CERT* discloses that dynamic content is filtered, and apparently does so regardless of whether the dynamic content is provided by the server, a user, or an outside source. In other words, *CERT* does not limit examinations for script constructs to any single element, let alone to only HTML elements where user input is introduced.[3]

Applicant also respectfully submits that the *Wheeler* reference fails to remedy the deficiencies of *CERT-Advisory* and *CERT*. Indeed, *Wheeler* discloses a general filtering subroutine to remove special characters, but does not disclose that the general subroutine is limited to a server applying it only to data derived from an outside source, as recited in combination with the other claim elements. (*See* §§ 4.10, 6.15-6.15.2.2).

For example, *Wheeler* expressly approves of the filtering method of *CERT-Advisory* and *CERT* and notes that to make a program safe, the "output must be filtered (so characters that can cause this problem are removed), encoded..., or validated." (§ 6.15.2, emphasis added). Such filtering, as described by *Wheeler* entails removing the special or "bad" characters, while leaving valid characters unaffected. (§ 6.15.2.2). Thus, in direct opposition to the pending claims in which examination occurs before rendering, *Wheeler* filters output after rendering. Furthermore, *Wheeler* discloses that special or bad characters are removed, but *Wheeler* fails to disclose that any characters or portions of a request that are not HTML elements where user input is introduced, are not examined. Indeed, any portion, whether static, dynamic, user input, server input, or of other nature appears to be filtered.

These deficiencies are further highlighted in the *Sanin* reference. In *Sanin* a web service security filter is disclosed and comprises a server-side plug-in that processes HTTP requests before any other Web service plug-in or application. (*Abstract*). In operation, an HTTP request

[3] Notably, the Office Action acknowledges that *CERT-Advisory* does not disclose determining whether a request includes examining only a second portion (corresponding to all data derived from an outside source). (Office Action, p. 9). The Office Action then turns to the *CERT* reference and, while stating that it discloses mitigating attacks by evaluating the incoming requests, does not even assert that it discloses or suggests limiting such consideration to only portions containing data derived from outside sources, let alone to only HTML elements where user input is introduced.

is processed by a filter before it reaches a Web service application. (¶ 22). The filter operates by parsing the HTTP requests into five categories of objects, and inspects the objects, category by category. (¶ 22). In particular, a method is disclosed that loads a group of pattern rules. (¶¶ 28-29). The incoming HTTP request is then parsed according to the objects and the group of pattern rules is applied to the objects. (¶¶ 30-31). If any substring included in the objects matches a predefined pattern, a rule action is taken (e.g., to reject the request as a bad request). (¶¶32-33). If none of the HTTP request objects matches any rule pattern, then the request is passed for further processing. (¶ 40).

Accordingly, in contrast to Applicant's claimed invention, in which a server examines only HTML elements where user input is introduced, *Sanin* discloses that all substrings in the objects are examined, regardless of whether they include data from a user, the server, or an outside source. Thus, the cited references, whether considered alone or in combination, not only fail to disclose each and every element as claimed by Applicant, but actually teach the opposite and teach that elements above and beyond user input HTML elements should be filtered and examined.

In view of the foregoing, Applicant respectfully submits that the other rejections to the claims are now moot and do not, therefore, need to be addressed individually at this time. It will be appreciated, however, that this should not be construed as Applicant acquiescing to any of the purported teachings or assertions made in the last action regarding the cited art or the pending application, including any official notice. Instead, Applicant reserves the right to challenge any of the purported teachings or assertions made in the last action at any appropriate time in the future, should the need arise. Furthermore, to the extent that the Examiner has relied on any Official Notice, explicitly or implicitly, Applicant specifically requests that the Examiner provide references supporting the teachings officially noticed, as well as the required motivation or suggestion to combine the relied upon notice with the other art of record.

In the event that the Examiner finds remaining impediment to a prompt allowance of this application that may be clarified through a telephone interview, the Examiner is requested to contact the undersigned attorney at (801) 533-9800.

Dated this 31st day of March, 2008.

Respectfully submitted,

RICK D. NYDEGGER
Registration No. 28,651
COLBY C. NUTTALL
Registration No. 58,146
Attorneys for Applicant
Customer No. 047973

RDN:CCN:gd
MS 373 - DRAFT AMENDE AFTER NON-FINAL (OA DATED 31OCT07) (2)